

Comparación de ejecución en lenguajes compilado, interpretado e híbrida

Integrantes:

Rosa Alejandra López Lizarazo - 20222020134

Jaiber Duvan Diaz León - 20222020011

Andres Nicolás Mendoza Silva- 20222020152

A lo largo de este trabajo hemos realizado una interesante comparativa entre 3 tipos de lenguajes diferentes, como lo son los compilados, interpretados e híbridos, usando C ++, Python y Java respectivamente. La comparación se realizó desarrollando 2 programas simples en cada uno de los lenguajes, el primer programa se trata de una calculadora que permitiera realizar operaciones básicas entre números de hasta 50 caracteres y el segundo programa por su parte se trató de un algoritmo para resolver Sudoku.

El desarrollo de estos programas no permite analizar el rendimiento de cada lenguaje de programación como también el tiempo que toma en ejecutarse dependiendo si el programa debe ser compilado, interpretado o se ejecuta de forma híbrida. De esta forma se nota la importancia de esta comparación, ya que cada tipo de lenguaje puede tener ventajas o desventajas en cada contexto y su naturaleza puede influir en el rendimiento del programa en situaciones reales.

Calculadoras:

Como se mencionó anteriormente las calculadoras debían soportar variables de hasta 50 dígitos, en este contexto es importante entender que para realizar operaciones con números tan grandes es necesario modificar la sintaxis y usar las bibliotecas de las que cada uno dispone.

Para el lenguaje C ++ fue necesario diseñar algoritmos con ayuda de la información en internet, debido a que no existe una función o biblioteca nativa del lenguaje que pudiera solucionar el problema. En este caso las operaciones se realizan caracterizando cada dígito por medio de cadenas y dando gestión a acarreo para obtener el número.

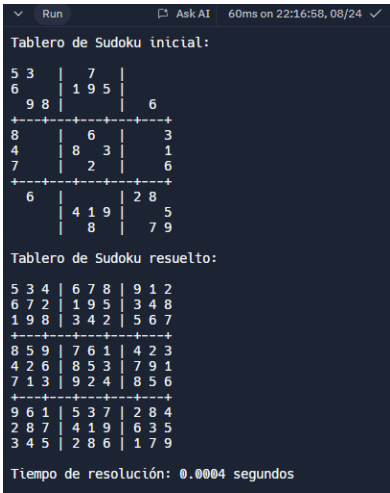
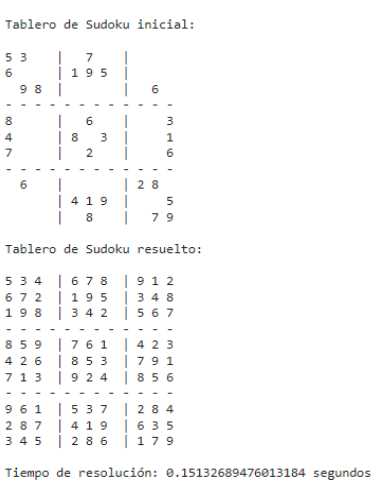
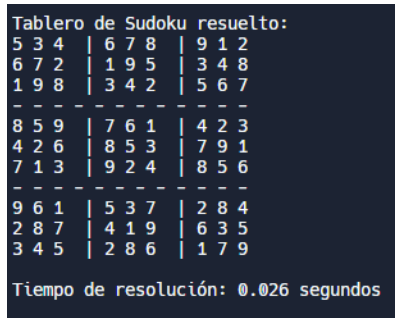
En Java hizo uso de la biblioteca “Java.math”, la cual incluye a la clase “BigInteger”, que está diseñada precisamente para manejar números enteros de grandes magnitudes de forma eficiente.

En el caso de Python no se tuvo que implementar alguna biblioteca o algoritmo y fue más sencillo gracias a que el tipo de dato “int” tiene una útil capacidad para el manejo de números enteros de grandes tamaños.

Entre los tres lenguajes usados para desarrollar el programa de calculadora, Python resultó ser el más rápido, a pesar de ser un lenguaje interpretado. Esto se debe a que Python utiliza de forma nativa un tipo de dato llamado int que soporta enteros de tamaño arbitrario. Por el contrario, tanto C++ como Java requieren el uso de bibliotecas especiales o el desarrollo de algoritmos para manejar números más grandes, lo cual a diferencia de Python no resulta ser tan óptimo a la hora de la ejecución del programa.

Compilado C++	Interpretado Python	Híbrido Java
<pre>Ingrese el número de la operación: 1 Ingrese el primer número: 98765432109876543210 Ingrese el segundo número: 123456789012345678901234567890 Resultado de la suma: 111111110111111110111111100 Tiempo de ejecución: 3.1488e-05 segundos</pre>	<pre>Ingrese el número de la operación: 1 Ingrese el primer número: 987654321098765432109876543210 Ingrese el segundo número: 123456789012345678901234567890 Resultado de la suma: 111111110111111110111111100 Tiempo de ejecución de la operación: 0.000049 segundos</pre>	<pre>Ingrese el primer número: 123456789012345678901234567890 Ingrese el segundo número: 987654321098765432109876543210 Seleccione la operación que desea realizar: 1. Suma 2. Resta 3. Multiplicación 4. División Ingrese el número de la operación: 1 Resultado de la suma: 111111110111111110111111100 Tiempo de ejecución: 21 ms</pre>

Sudokus:

Compilado C++	Interpretado Python	Híbrido Java
 <p>C++ es un lenguaje compilado que genera código máquina optimizado para la arquitectura en la que se ejecuta, lo que permite tiempos de ejecución mucho más rápidos. Además, C++</p>	 <p>Python, al ser un lenguaje interpretado, introduce una sobrecarga considerable al momento de ejecutar el código. Además, Python realiza más operaciones en tiempo de ejecución, como la</p>	 <p>Java, un lenguaje compilado a bytecode que se ejecuta en una máquina virtual (JVM), demostró un rendimiento que podría estar entre C++ y Python, aunque en este caso específico, el tiempo es extremadamente bajo, lo que sugiere una optimización en la JVM y la eficiencia de los algoritmos.</p> <p>Similar en estructura a C++,</p>

gestiona la memoria de manera más eficiente, lo que reduce la sobrecarga durante la ejecución.	gestión automática de memoria, lo que puede aumentar el tiempo de ejecución.	pero con la sobrecarga de la JVM. Sin embargo, aún muestra un rendimiento excelente con un tiempo de 0.026 segundos, cercano al de C ++.
--	--	--

Para tareas que requieren un rendimiento muy alto, como la resolución de problemas computacionalmente intensivos, C ++ es claramente la mejor opción. La diferencia de tiempo es significativa, y aunque Python ofrece una mayor facilidad de uso y desarrollo más rápido, no puede competir con C ++ en términos de velocidad de ejecución en este contexto.

El resultado en Java es muy competitivo, con un tiempo de 0.026 segundos para resolver el Sudoku, que es comparable al tiempo que obtendremos en C ++. Esto demuestra que Java, aunque más abstracto y gestionado que C ++, aún puede ofrecer tiempos de ejecución muy eficientes para aplicaciones críticas en rendimiento, como lo es la resolución de un Sudoku utilizando backtracking.

Este resultado refuerza la idea de que, aunque C ++ es tradicionalmente visto como la opción más rápida, Java puede ofrecer tiempos de ejecución comparables dependiendo de la optimización de la JVM y la implementación específica del algoritmo. Python, aunque más lento, sigue siendo útil en situaciones donde el tiempo de desarrollo es más crítico que el tiempo de ejecución.